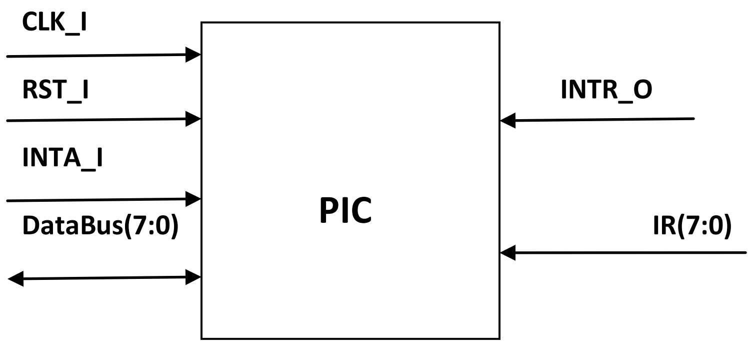
DOCUMENTATION

“pic” is a soft core programmable interrupt controller which can be used as an interface between peripheral interrupt lines and processor IRQ line. One of the popular PIC available in market is Intel 8259. This core is not compatible with 8259. The core was designed based on my ideas of how a PIC operates and its requirements. The first version is a really basic core which can take 8 interrupts as input. The interrupt detecting methods currently supported are polling method and fixed priority method.

PIC Core Top entity:

The core can be considered as a black box with the following inputs and outputs:



IO ports:-

CLK_I : Input clock at which the system works.

RST_I : Reset input to the system.

INTA_I : Interrupt acknowledge from the processor.It is also used for acknowledging any data transfer between the processor and pic core.

INTR_O : Interrupt request output. This is an output from the core which is connect to the hardware input pin of the processor.

IR(7 : 0) : Interrupt request inputs from upto 8 peripherals.

DataBus(7:0) : It is a inout port which is used for communication between processor and PIC. The different modes and interrupt priorities can be selected by writing data into this bus. Also the PIC writes data into this bus when a new interrupt occurs.

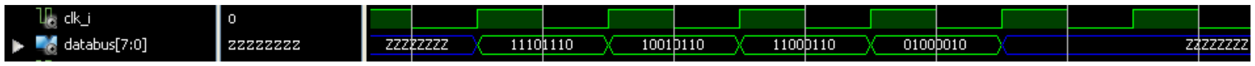
Two different methods are used to handle interrupts in this core:

1)**Polling method** – One by one each interrupt line is checked to see whether an interrupt has occurred. So the worst delay in informing the processor about any interrupt in this mode is 8 clock cycles. This mode assumes that all the interrupts have the same priority.

2)**Fixed priority method** – In this method all the eight interrupts are given a priority number each one different from another. The small the assigned number is the highest is its priority. If more than one interrupt occurs then the interrupt with the highest priority gets preference over others.

Before changing the operating mode you have to assert the RST_I signal for one clock cycle. After this you can apply the inputs as shown in the following waveform:

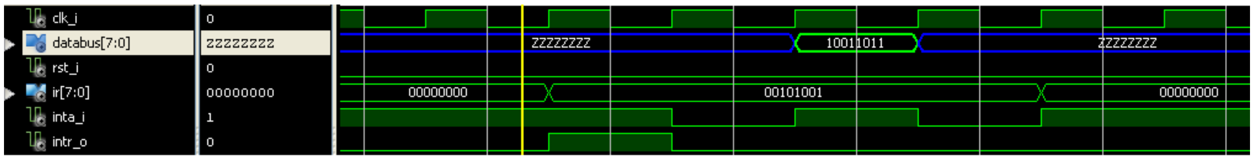
a) For polling method simple send "ZZZZZZ**01**" for one clock cycle.
b) For fixed priority method check the waveform:



Note that here the LSB 2 bits should be always "10". This is for telling the PIC that operating mode is "fixed priority". The MSB bits are the interrupt index in their priority order. In the wave form shown above we are transmitting "11101110","10010110","11000110" and "01000010". So interrupt number 7 has priority number 0 which means it has the highest priority. And the rest of the interrupts in their descending order of priority are 3,4,5,6,1,2,0.

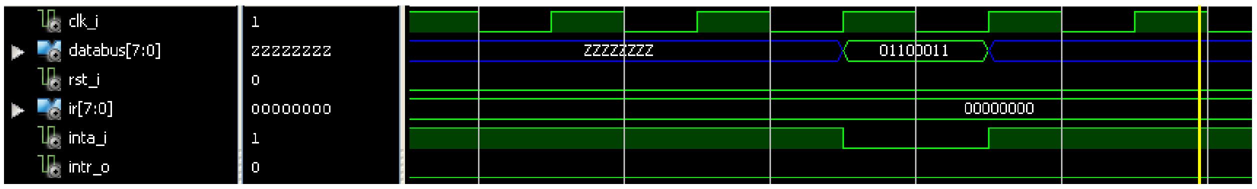
When an interrupt occurs the PIC core will assert a high on the INTR_O output. Once this signal is received, processor has to send a acknowledgement signal INTA_I to the core.On this signal assertion PIC core will send the detail about the interrupt along with some matching bits to the processor. The matching bits for polling type mode is "01011" and for fixed priority type is "10011".The LSB 3 bits will contain the index of the interrupt occurred. On receipt of these bits processor has to send out a INTA_I signal again.

The below waveform is for fixed priority type.



At the end of interrupt service routine(ISR) processor should send a acknowledge signal, INTA_I along with some data on the databus(7:0). The MSB 5 bits of the databus for polling type is “10100” and for fixed priority type is “01100”. The LSB 3 bits contain the index of interrupt whose ISR is finished.

The below waveform is for fixed priority type:



Improvements for future versions of this PIC:

1)Cascading of PICs to support upto 64 interrupts.

2)More modes of operation.